

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Enclosure: a/s

ATTORNEY DOCKET No. 2557-000211  
HANESS, DICKEY & PERCE  
703-668-8000



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 10-2003-0011210  
Application Number

출원년월일 : 2003년 02월 22일  
Date of Application FEB 22, 2003

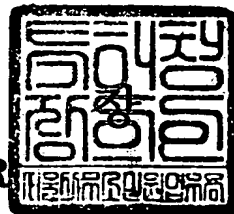
출원인 : 삼성전자주식회사  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003      년      07      월      15      일

특      허      청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0023
【제출일자】	2003.02.22
【국제특허분류】	G06F
【발명의 명칭】	블록 캐리 전파 즉시 합산 값을 출력하는 한 위상내 자체 동기 캐리 룩어헤드 애더 및 그 합산 방법
【발명의 영문명칭】	One-phase self-timed carry lookahead adder providing for summation value outputting at block carry propagation and summation method thereof
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	정상빈
【대리인코드】	9-1998-000541-1
【포괄위임등록번호】	2003-003437-4
【발명자】	
【성명의 국문표기】	최창준
【성명의 영문표기】	CHOI, Chang Jun
【주민등록번호】	710721-1411113
【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 황골마을아파트 155-1802
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이영필 (인) 대리인 정상빈 (인)

**【수수료】**

【기본출원료】 20 면 29,000 원

【가산출원료】 23 면 23,000 원

【우선권주장료】 0 건 0 원

【심사청구료】 20 항 749,000 원

【합계】 801,000 원

**【첨부서류】**

1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

블록 캐리 전파 즉시 합산 값을 출력하는 한 위상내 자체 동기 캐리 록어헤드 애더 및 그 합산 방법이 개시된다. 상기 캐리 록어헤드 애더는, 먼저, 캐리 생성/전파부가 N 비트의 바이너리 값으로 입력되는 가산수와 피가산수를 M 블록들로 분별하여, 상기 블록들 각각에서 비트별로 제1 논리 조합하여 캐리 전파 비트 값 및 캐리 소모 비트 값을 계산한다. 블록 캐리부는 블록 캐리들을 생성하고, 맨체스터 캐리 체인으로 구성되는 비트 캐리부는 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 비트별 캐리들을 계산한다. 이에 따라, 블록 캐리 선택부가 블록 캐리 발생 정보를 생성하면, 블록 캐리 발생 정보에 동기화된 합산 값 선택부는 상기 블록 캐리 발생 정보에 대응하는 상기 비트별 캐리들을 선택하고, 선택된 상기 비트별 캐리들에 비트별 상기 캐리 전파 비트 값을 비트별로 합산하여 최종 합산 결과를 출력한다. 따라서, 바이너리 값을 가지는 가산수와 피가산수에 대하여 클럭 신호의 한 페이즈 내에 고속 합산을 수행 할 수 있는 효과가 있다.

**【대표도】**

도 1

**【명세서】****【발명의 명칭】**

블록 캐리 전파 즉시 합산 값을 출력하는 한 위상내 자체 동기 캐리 룩어헤드 애더 및 그 합산 방법{One-phase self-timed carry lookahead adder providing for summation value outputting at block carry propagation and summation method thereof}

**【도면의 간단한 설명】**

본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 간단한 설명이 제공된다.

도 1은 본 발명의 일실시예에 따른 캐리 룩어헤드 애더(CLA)의 블록도이다.

도 2는 도 1을 구체적으로 보여주는 다른 블록도이다.

도 3은 도 1의 캐리 생성/전파부(110)의 구체적인 회로도이다.

도 4는 도 1의 블록 캐리부(120)의 구체적인 회로도이다.

도 5a 및 도 5b는 도 1의 비트 캐리부(130)의 구체적인 회로도이다.

도 6은 도 1의 블록 캐리 선택부(140)의 구체적인 회로도이다.

도 7은 도 1의 합산 값 선택부(150)의 구체적인 회로도이다.

도 8은 도 3 내지 도 7을 결합한 전체 회로에 대한 시뮬레이션 파형도이다.

## 【발명의 상세한 설명】

## 【발명의 목적】

## 【발명이 속하는 기술분야 및 그 분야의 종래기술】

<10> 본 발명은 캐리 룩어헤드 애더(Carry Lookahead Adder)(이하 "CLA"로 약칭함)에 관한 것으로, 특히 맨체스터(Manchester) 캐리 체인을 기초로 하는 캐리 룩어헤드 애더(CLA)에 관한 것이다.

<11> 일반적인 애더 회로의 딜레이는 캐리 전파(carry-propagation) 경로(path)에 의해 결정된다. 캐리 전파 경로(path)를 줄여 동작 속도를 증가시키기 위하여, 맨체스터 캐리 체인을 사용하는 종래의 캐리 룩어헤드 애더(CLA)는, 바이너리(binary) 값을 가지는 합산될 두 연산자, 즉 가산수(addend)와 피가산수(augend)에 대한, 캐리 전파(propagation) 비트 및 캐리 생성(generation) 비트를 생성하여, 블록 캐리에 의하여 선택되는 비트 캐리로부터 최종 합산 값을 계산한다. 이와 같은 종래의 캐리 룩어헤드 애더(CLA)에 대하여, 미국 특허, "US5,508,952A" 또는 일본 공개 특허, "JP1993-61643"에 잘 나타나 있다.

<12> 즉, 종래의 캐리 룩어헤드 애더(CLA)는, 합산 동작의 시작을 알리는 소정 클럭 신호의 라이징 트랜지션(rising transition), 즉, 액티브시에 발생하는 캐리 전파 비트 및 캐리 생성 비트를 이용하여 블록 캐리와 비트 캐리들을 계산한다. 그런데, 종래의 캐리 룩어헤드 애더(CLA)에서는, 상기 클럭 신호의 액티브 기간 내의 동일 페이즈(phase) 내에서 최종 합산 값을 계산하지 않고, 상기 클럭 신호의 폴링 에지(falling edge)에 동기된 소정 인에이블(enable) 신호가 액티브될 때, 즉, 상기 클럭 신호의 액티브 상태 다음

의 페이지에서, 감지 증폭 플립플롭(sense amp. F/F)이 구동되어 상기 블록 캐리에 의하여 선택되는 비트 캐리로부터 최종 합산 값이 계산된다.

<13> 따라서, 이와 같은 종래의 캐리 룩어헤드 애더(CLA)에서는, 블록 캐리의 생성 시점이 가산수와 피가산수의 값에 따라 달라지기 때문에, 전체 비트에 대한 정상적인 합산 동작을 보장하기 위하여 클럭 신호의 타이밍 마진(timing margin)이 충분히 고려되어야 한다. 즉, 상기 클럭 신호가 라이징 트랜지션(rising transition)하여 액티브 상태로 된 기간 동안에 생성되는 캐리의 전파가 진행되어도, 감지 증폭 플립플롭(sense amp. F/F)이 동작하기 위해서는 상기 클럭 신호의 다음 페이지까지 기다려야 하는 시간적 손실이 발생하는 문제점이 있다.

#### 【발명이 이루고자 하는 기술적 과제】

<14> 따라서, 본 발명이 이루고자하는 기술적 과제는, 맨체스터 캐리 체인을 사용한 캐리 룩어헤드 애더(CLA)에서, 감지 증폭 플립플롭(sense amp. F/F)을 액티브시키는 인에이블 신호를 기다리는 시간적 손실을 없애어, 블록 캐리가 생성되어 전파되는 즉시, 블록 캐리에 의하여 선택되는 비트 캐리로부터 최종 합산 값을 계산하는 소정 동기화기(synchronizer)를 가지는 한 위상 내 자체 동기 캐리 룩어헤드 애더(CLA) 및 그 합산 방법을 제공하는 데 있다.

#### 【발명의 구성 및 작용】

<15> 상기의 기술적 과제를 달성하기 위한 본 발명에 따른 캐리 룩어헤드 애더(CLA)는, 캐리 생성/전파부, 블록 캐리부, 비트 캐리부, 블록 캐리 선택부, 및 합산 값 선택부를 구비한다.

- <16>      상기 캐리 생성/전파부는  $N$  비트의 바이너리 값으로 입력되는 가산수와 피가산수를  $M$  블록들로 분별하여, 상기 블록들 각각에서 비트별로 제1 논리 조합하여 캐리 전파 비트 값(PH) 및 캐리 소모 비트 값(KH)을 계산한다.
- <17>      상기 블록 캐리부는 상기 블록들 각각에서, 비트별 상기 캐리 소모 비트 값(KH)에 대한 제2 논리 조합에 의하여 하나 이상의 제2 논리 상태가 존재하면 제2 논리 상태로 되는  $K0$ , 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는  $K4$ , 및 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는  $K8$ 로 이루어지는 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ )을 생성한다.
- <18>      상기 비트 캐리부는 상기 블록들 각각에서, 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 상기 캐리 전파 비트 값(PH) 및 상기 캐리 소모 비트 값(KH)에 대한 제5 논리 조합으로 비트별 캐리들( $C0$ , 및  $C1$ )을 맨체스터 캐리 체인에 의하여 계산한다.
- <19>      상기 블록 캐리 선택부는 상기 블록들 각각에서, 상기 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ ) 및 하위 비트 블록으로부터 전파되는 캐리 값에 대한 소정 논리 조합에 의하여, 제1 논리 상태 또는 제2 논리 상태를 가지며, 어떠한 클럭 신호에 의하여도 출력을 제어 받지 않는 블록 캐리 발생 정보(CINH, 및 CINL)를 생성하는 동기화기(synchronizer)이다. 여기서, 블록 캐리 선택부는 블록 캐리 발생 정보(CINH, 및 CINL)를 생성하고, 이에 따라

동기화(synchronizing)되는 상기 합산 값 선택부는, 상기 블록 캐리들(K0, K4, 및 K8)이 생성되는 즉시, 같은 페이즈(phase)의 클럭 신호 내에서 최종 합산 결과(SUM)를 출력한다.

<20> 즉, 상기 합산 값 선택부는 상기 블록 캐리 발생 정보(CINH, 및 CINL)에 대응하는 상기 비트별 캐리(C0, 또는 C1)을 선택하고, 선택된 상기 비트별 캐리(C0, 또는 C1)에 비트별 상기 캐리 전과 비트 값(PH)을 비트별로 합산하여 최종 캐리(CM)를 포함하는 최종 합산 결과(SUM)를 출력한다.

<21> 여기서, 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 서로 반전된 논리 상태를 가지는 한 쌍의 디지털 정보이며, 상기 블록 캐리들(K0, K4, 및 K8) 모두 제1 논리 상태에서 블록 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택되도록 하는 것을 특징으로 한다.

<22> 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 상기 하위 비트 블록으로부터 전파되는 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택되도록 하는 것을 특징으로 한다.

<23> 상기 하위 비트 블록은, 상기 합산이 이루어지는 블록의 바로 이전 하위 비트 블록, 및 상기 합산이 이루어지는 블록의 2번째 이전 하위 비트 블록을 포함하는 것을 특징으로 한다.

<24> 상기 제1 논리 조합은, 상기 가산수와 상기 피가산수에 대한 비트별 XOR 로직에 의하여 상기 캐리 전과 비트 값(PH)을 계산하는 로직, 및 상기 가산수와 상기 피가산수에

대한 비트별 NOR 로직에 의하여 상기 캐리 소모 비트 값(KH)을 계산하는 로직인 것을 특징으로 한다.

<25> 상기 제2 논리 조합은, 비트별 상기 캐리 소모 비트 값(KH)에 대한 OR 로직인 것을 특징으로 한다.

<26> 상기 제3 논리 조합은, 논리식,

<27>  $K4 =$

<28>  $PH<\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{N/(2M)+1\}>PH<N/(2M)>$

<29>  $\{KH<\{N/(2M)-1\}> +$

<30>  $PH<\{N/(2M)-1\}>KH<\{N/(2M)-2\}> +$

<31>  $PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>KH<\{N/(2M)-3\}> +$

<32>  $\dots + PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>\dots PH<1>KH<0> +$

<33>  $PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>\dots PH<1>PH<0>\}$

<34> (여기서, K4는 하위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH<>$ 는 캐리 전파 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

<35> 을 만족하는 것을 특징으로 한다.

<36> 상기 제4 논리 조합은, 논리식,

<37>  $K8 =$

<38>  $KH<\{(N/M)-1\}> +$

<39>  $PH<\{(N/M)-1\}>KH<\{(N/M)-2\}> +$

<40>  $\dots + PH<\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{(N/(2M)+2)\}>KH<\{N/(2M)+1\}> +$

- <41>  $PH\langle\{(N/M)-1\}\rangle PH\langle\{(N/M)-2\}\rangle \dots PH\langle\{(N/(2M)+1)\}\rangle KH\langle N/(2M)\rangle$
- <42> (여기서, K8은 상위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH\langle\rangle$ 는 캐리 전파 비트 값,  $KH\langle\rangle$ 는 캐리 소모 비트 값)
- <43> 을 만족하는 것을 특징으로 한다.
- <44> 상기 제5 논리 조합은, 논리식,
- <45>  $C0\langle i\rangle = KH\langle i-1\rangle + PH\langle i-1\rangle C0\langle i-1\rangle$
- <46>  $C1\langle i\rangle = KH\langle i-1\rangle + PH\langle i-1\rangle C1\langle i-1\rangle$
- <47> (여기서,  $i$ 는 0부터  $(N/M-1)$ 까지,  $C1\langle i\rangle$ 은 첫 번째 비트에 캐리가 있는 경우의 비트별 캐리들,  $C0\langle i\rangle$ 은 첫 번째 비트에 캐리가 없는 경우의 비트별 캐리들,  $PH\langle\rangle$ 는 캐리 전파 비트 값,  $KH\langle\rangle$ 는 캐리 소모 비트 값)
- <48> 을 만족하는 것을 특징으로 한다.
- <49> 상기 합산은, 상기 비트별 캐리들( $C0$ , 및  $C1$ )과 비트별 상기 캐리 전파 비트 값( $PH$ )을 비트별로 XOR 로직 처리하여 계산하는 것을 특징으로 한다.
- <50> 상기의 다른 기술적 과제를 달성하기 위한 본 발명에 따른 캐리 룩어헤드 애더(CLA)의 합산 방법은,  $N$  비트의 바이너리 값으로 입력되는 가산수와 피가산수를  $M$  블록들로 분별하여 계산되는 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ )에 대응하여, 상기 블록들 각각의 맨체스터 캐리 체인에서 발생하는 비트 캐리들이 선택되도록 하는 캐리 룩어헤드 애더(CLA)의 합산 방법에 있어서, 다음과 같은 단계를 구비한다.
- <51> 즉, 본 발명에 따른 캐리 룩어헤드 애더(CLA)의 합산 방법은, 먼저, 상기 캐리 룩어헤드 애더(CLA)가 상기 블록들 각각에서 상기 가산수와 상기 피가산수의 비트별로 제1

논리 조합하여 캐리 전파 비트 값(PH) 및 캐리 소모 비트 값(KH)을 계산한다. 다음에, 상기 캐리 룩어헤드 애더(CLA)는, 상기 블록들 각각에서, 비트별 상기 캐리 소모 비트 값(KH)에 대한 제2 논리 조합에 의하여 하나 이상의 제2 논리 상태가 존재하면 제2 논리 상태로 되는 K0, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는 K4, 및 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는 K8로 이루어지는 블록 캐리들(K0, K4, 및 K8)을 생성한다.

<52>        상기 캐리 룩어헤드 애더(CLA)는, 상기 블록들 각각에서, 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 상기 캐리 전파 비트 값(PH) 및 상기 캐리 소모 비트 값(KH)에 대한 제5 논리 조합으로 상기 비트별 캐리들(C0, 및 C1)을 상기 맨체스터 캐리 체인에 의하여 계산한다. 상기 캐리 룩어헤드 애더(CLA)는, 소정 동기화기(synchronizer)에 의하여, 상기 블록들 각각에서, 상기 블록 캐리들(K0, K4, 및 K8) 및 하위 비트 블록으로부터 전파되는 캐리 값에 대한 소정 논리 조합으로 제1 논리 상태 또는 제2 논리 상태를 가지며, 어떠한 클럭 신호에 의하여도 출력을 제어 받지 않는 블록 캐리 발생 정보(CINH, 및 CINL)를 생성한다.

<53>        이에 따라, 상기 캐리 룩어헤드 애더(CLA)는, 상기 블록 캐리 발생 정보(CINH, 및 CINL)에 동기시켜, 이에 대응하는 상기 비트별 캐리(C0, 또는 C1)을 선택하고, 선택된 상기 비트별 캐리(C0, 또는 C1)에 비트별 상기 캐리 전파 비트 값(PH)을 비트별로 합산하여 최종 캐리(CM)를 포함하는 최종 합산 결과(SUM)를 출력한다.

- <54> 여기서, 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 서로 반전된 논리 상태를 가지는 한 쌍의 디지털 정보이며, 상기 블록 캐리들(K0, K4, 및 K8) 모두 제1 논리 상태에서 블록 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택되도록 하는 것을 특징으로 한다.
- <55> 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 상기 하위 비트 블록으로부터 전파되는 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택되도록 하는 것을 특징으로 한다.
- <56> 상기 하위 비트 블록은, 상기 합산이 이루어지는 블록의 바로 이전 하위 비트 블록, 및 상기 합산이 이루어지는 블록의 2번째 이전 하위 비트 블록을 포함하는 것을 특징으로 한다.
- <57> 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.
- <58> 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 각 도면에 제시된 동일한 참조부호는 동일한 부재를 나타낸다.
- <59> 도 1은 본 발명의 일실시예에 따른 캐리 록어헤드 애더(CLA)의 블록도이고, 도 2는 도 1을 구체적으로 보여주는 다른 블록도이다.
- <60> 도 1 및 도 2를 참조하면, 본 발명의 일실시예에 따른 캐리 록어헤드 애더(CLA)는, 캐리 생성/전파부(carry generation/propagation unit)(110), 블록 캐리부(block carry

unit)(120), 비트 캐리부(bit carry unit)(130), 블록 캐리 선택부(block carry selection unit)(140), 및 합산 값 선택부(summation selection unit)(150)를 구비한다.

<61>      상기 캐리 생성/전파부(110)는 N 비트의 바이너리 값으로 입력되는 가산수 (addend)(A)와 피가산수(augend)(B)를 M 블록들로 분별하여, 상기 블록들 각각에서 비트 별로 제1 논리 조합하여 캐리 전파 비트 값(PH) 및 캐리 소모 비트 값(KH)을 계산한다. 도 2에서는, 상기 가산수(A)와 피가산수(B)를 32 비트( $N=32$ )로 가정하였고, 이들이 4 블록( $M=4$ )으로 분별되는 것으로 가정하였다.

<62>      여기서, 상기 제1 논리 조합은, [표 1]과 같이, 상기 가산수와 상기 피가산수에 대한 비트별 XOR 로직에 의하여 상기 캐리 전파 비트 값(PH)을 계산하는 로직, 및 상기 가산수와 상기 피가산수에 대한 비트별 NOR 로직에 의하여 상기 캐리 소모 비트 값(KH)을 계산하는 로직이다. [표 1]에서, "0"은 제1 논리 상태, 즉, 논리 로우 상태이고, "1"은 제2 논리 상태, 즉, 논리 하이 상태이다. 또한, i는 상기 블록 내에 존재하는 모든 비트를 표시하는 것으로, 0부터  $\{(N/M)-1\}$ 까지의 비트를 표시하고, 특히 도 2에서는 0부터 7까지에 해당한다. 즉, 도 2에서 N은 32이고, M은 4이며,  $N/(2M)$ 은 4에 해당한다.

<63>      [표 1]

<64>

A<i>	B<i>	PH<i>	KH<i>
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	0

<65>      상기 블록 캐리부(120)는 상기 블록들 각각에서, 비트별 상기 캐리 소모 비트 값 (KH)에 대한 제2 논리 조합에 의하여 하나 이상의 제2 논리 상태가 존재하면 제2 논리 상태로 되는 K0, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값

(KH)에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들(도 2의 각 블록에서 하위 4비트)에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는 K4, 및 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들(도 2의 각 블록에서 상위 4비트)에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는 K8로 이루어지는 블록 캐리들(K0, K4, 및 K8)을 생성한다.

<66> 여기서, 상기 제2 논리 조합은, 비트별 상기 캐리 소모 비트 값(KH)에 대한 OR 로직으로서, [논리식 1]과 같다. [논리식 1]에서, K0은 한 블록 내의 비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리이다. 또한, 캐리 소모 비트 값(KH) 중 하나 이상의 제2 논리 상태가 존재하면, 아래에서 기술할 비트 캐리부(130)에서 발생하는 캐리가 맨체스터 캐리 체인 동작 원리에 의하여 소모되도록 한다(즉, 킬(kill) 시킨다).

<67> [논리식 1]

<68> 
$$K0 = KH<0> + KH<1> + \dots + KH<6> + KH<N/(2M)>$$

<69> (여기서, K0은 비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리, KH<>는 캐리 소모 비트 값)

<70> 상기 제3 논리 조합은, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들(도 2의 각 블록에서 하위 4비트)에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는 로직으로서, [논리식 2]를 만족한다. [논리식 2]에서, K4는 하위  $N/(2M)$  비트들에 킬텀(kill term), 즉, 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이

하나 이상 존재하는지 체크하는 블록 캐리이다. 도 2에 대하여, [논리식 2]는 [논리식 3]과 같이 된다.

<71> [논리식 2]

<72>  $K_4 =$

<73>  $PH\langle\{N/(2M)-1\}\rangle PH\langle\{N/(2M)-2\}\rangle \dots PH\langle\{N/(2M)+1\}\rangle PH\langle N/(2M)\rangle$

<74>  $\{KH\langle\{N/(2M)-1\}\rangle\} +$

<75>  $PH\langle\{N/(2M)-1\}\rangle KH\langle\{N/(2M)-2\}\rangle +$

<76>  $PH\langle\{N/(2M)-1\}\rangle PH\langle\{N/(2M)-2\}\rangle KH\langle\{N/(2M)-3\}\rangle +$

<77>  $\dots + PH\langle\{N/(2M)-1\}\rangle PH\langle\{N/(2M)-2\}\rangle \dots PH\langle 1 \rangle KH\langle 0 \rangle +$

<78>  $PH\langle\{N/(2M)-1\}\rangle PH\langle\{N/(2M)-2\}\rangle \dots PH\langle 1 \rangle PH\langle 0 \rangle\}$

<79> (여기서,  $K_4$ 는 하위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH\langle \rangle$ 는 캐리 전파 비트 값,  $KH\langle \rangle$ 는 캐리 소모 비트 값)

<80> [논리식 3]

<81>  $K_4 = PH\langle 7 \rangle PH\langle 6 \rangle PH\langle 5 \rangle PH\langle 4 \rangle \{KH\langle 3 \rangle + PH\langle 3 \rangle KH\langle 2 \rangle + PH\langle 3 \rangle PH\langle 2 \rangle KH\langle 1 \rangle +$

<82>  $PH\langle 3 \rangle PH\langle 2 \rangle PH\langle 1 \rangle KH\langle 0 \rangle + PH\langle 3 \rangle PH\langle 2 \rangle PH\langle 1 \rangle PH\langle 0 \rangle\}$

<83> 상기 제4 논리 조합은, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들(도 2의 각 블록에서 상위 4비트)에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 되는 로직으로서, [논리식 4]를 만족한다. [논리식 4]에서,  $K_8$ 은 상위  $N/(2M)$  비트들에 킬텀(kill term), 즉, 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이

하나 이상 존재하는지 체크하는 블록 캐리이다. 도 2에 대하여, [논리식 4]는 [논리식 5]와 같이 된다.

<84> [논리식 4]

<85>  $K8 =$

<86>  $KH\langle\{(N/M)-1\}\rangle +$

<87>  $PH\langle\{(N/M)-1\}\rangle KH\langle\{(N/M)-2\}\rangle +$

<88>  $\dots + PH\langle\{(N/M)-1\}\rangle PH\langle\{(N/M)-2\}\rangle \dots PH\langle\{(N/(2M)+2)\}\rangle KH\langle\{N/(2M)+1\}\rangle +$

<89>  $PH\langle\{(N/M)-1\}\rangle PH\langle\{(N/M)-2\}\rangle \dots PH\langle\{(N/(2M)+1)\}\rangle KH\langle N/(2M)\rangle$

<90> (여기서, K8은 상위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH\langle\rangle$ 는 캐리 전파 비트 값,  $KH\langle\rangle$ 는 캐리 소모 비트 값)

<91> [논리식 5]

<92>  $K8 = KH\langle 7 \rangle + PH\langle 7 \rangle KH\langle 6 \rangle + PH\langle 7 \rangle PH\langle 6 \rangle KH\langle 5 \rangle + PH\langle 7 \rangle PH\langle 6 \rangle PH\langle 5 \rangle KH\langle 4 \rangle$

<93> 상기 비트 캐리부(130)는 상기 블록들 각각에서, 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 상기 캐리 전파 비트 값(PH) 및 상기 캐리 소모 비트 값(KH)에 대한 제5 논리 조합으로 비트별 캐리들(C0, 및 C1)을 맨체스터 캐리 체인에 의하여 계산한다. 맨체스터 캐리 체인은 체인에 인가되는 캐리 소모 비트 값(KH) 중 하나 이상의 제2 논리 상태가 존재하면, 발생하는 캐리가 소모되도록 하는(즉, 킬(kill)시키는) 회로로서, 이것에 대하여는 미국 특허, "US5,508,952A" 또는 일본 공개 특허, "JP1993-61643"에 잘 나타나 있다.

<94>       상기 제5 논리 조합은, [논리식 6]을 만족하는 로직이다. [논리식 6]에서,  $C1<i>$ 는 첫 번째 비트에 캐리가 있는 경우에 해당하고,  $C0<i>$ 는 첫 번째 비트에 캐리가 없는 경우에 해당한다.

<95> [논리식 6]

<96>        $C0<i> = KH<i-1> + PH<i-1>C0<i-1>$

<97>        $C1<i> = KH<i-1> + PH<i-1>C1<i-1>$

<98>       (여기서,  $i$ 는 0부터  $(N/M-1)$ 까지,  $C1<i>$ 은 첫 번째 비트에 캐리가 있는 경우의 비트별 캐리들,  $C0<i>$ 은 첫 번째 비트에 캐리가 없는 경우의 비트별 캐리들,  $PH<>$ 는 캐리 전과 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

<99>       상기 블록 캐리 선택부(140)는 상기 블록들 각각에서, 상기 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ ) 및 하위 비트 블록으로부터 전파되는 캐리 값에 대한 소정 논리 조합에 의하여, 제1 논리 상태 또는 제2 논리 상태를 가지며, 어떠한 클럭 신호에 의하여도 출력을 제어 받지 않는 블록 캐리 발생 정보( $CINH$ , 및  $CINL$ )를 생성하는 동기화기(synchronizer)이다

<100>       여기서, 상기 블록 캐리 발생 정보( $CINH$ , 및  $CINL$ )는, 서로 반전된 논리 상태를 가지는 한 쌍의 디지털 정보이며, 상기 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ ) 모두 제1 논리 상태로서 블록 캐리가 있는 경우에, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리( $C1$ )가 선택되도록 만들어진다. 또한, 상기 블록 캐리 발생 정보( $CINH$ , 및  $CINL$ )는, 상기 하위 비트 블록으로부터 전파되는 캐리가 있는 경우

에, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택되도록 만들어진다.

<101>       상기 하위 비트 블록은, 도 2에 도시된 바와 같이, 상기 합산이 이루어지는 블록의 바로 이전 하위 비트 블록, 및 상기 합산이 이루어지는 블록의 2번째 이전 하위 비트 블록을 포함한다. 즉, 하위 블록에서의 캐리 전파는 바로 이전 하위 비트 블록 또는 2번째 이전 하위 비트 블록에서 발생할 수 있고, 2번째 이전 하위 비트 블록에서 캐리가 전파되는 것은, 바로 이전 하위 비트 블록과 2번째 이전 하위 비트 블록 모두에서 캐리가 발생된 경우에 발생한다.

<102>       마찬가지로, 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 상기 블록 캐리들(K0, K4, 및 K8) 중 어느 하나가 제2 논리 상태로서 블록 캐리가 없는 경우에, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 없는 경우의 상기 비트별 캐리(C0)가 선택되도록 만들어진다. 또한, 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 상기 하위 비트 블록으로부터 전파되는 캐리가 없는 경우에, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 없는 경우의 상기 비트별 캐리(C0)가 선택되도록 만들어진다.

<103>       여기서, 블록 캐리 선택부(140)는 블록 캐리 발생 정보(CINH, 및 CINL)를 생성하고, 이에 따라 동기화(synchronizing)되는 상기 합산 값 선택부(150)는, 상기 블록 캐리들(K0, K4, 및 K8)이 생성되는 즉시, 같은 페이즈(phase)의 클럭 신호 내에서 최종 합산 결과(SUM)를 출력한다.

<104>       즉, 상기 합산 값 선택부(150)는 상기 블록 캐리 발생 정보(CINH, 및 CINL)에 대응하는 상기 비트별 캐리(C0, 또는 C1)를 선택하고, 선택된 상기 비트별 캐리(C0, 또는 C1)에 비트별 상기 캐리 전파 비트 값(PH)을 비트별로 합산하여 최종 캐리(CM)를 포함하

는 최종 합산 결과(SUM)를 출력한다. 여기서, 상기 합산은, [표 2]와 같이, 상기 비트별 캐리들(C0, 및 C1)과 비트별 상기 캐리 전파 비트 값(PH)을 비트별로 XOR 로직 처리함으로써 계산된다. [표 2]에서, "0"은 제1 논리 상태, 즉, 논리 로우 상태이고, "1"은 제2 논리 상태, 즉, 논리 하이 상태이다. 또한, i는 상기 블록 내에 존재하는 모든 비트를 표시하는 것으로, 0부터 {(N/M)-1}까지의 비트를 표시하고, 특히 도 2에서는 0부터 7까지에 해당한다.

<105> [표 2]

PH<i>	C0<i>	C1<i>	PH<i> + C0<i>	PH<i> + C1<i>
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	1	0	0

<107> 이와 같은 본 발명의 일실시예에 따른 캐리 록어헤드 애더(CLA)의 동작을 도 3 내지 도 7에 도시된 구체적인 회로도에 의하여 좀더 자세히 설명한다.

<108> 도 3은 도 1의 캐리 생성/전파부(110)의 구체적인 회로도이다.

<109> 도 3을 참조하면, 캐리 생성/전파부(110)는 N 비트의 바이너리 값으로 입력되는 가산수(A)와 피가산수(B)를 입력받아, 합산 시작을 알리는 클럭 신호(ACLK)가 액티브될 때, [표 1]과 같은 제1 논리 조합, 즉, 상기 가산수와 상기 피가산수에 대한 비트별 XOR 로직에 의하여 상기 캐리 전파 비트 값(PH)을 계산하는 로직, 및 상기 가산수와 상기 피가산수에 대한 비트별 NOR 로직 각각에 의하여, 비트별로 캐리 전파 비트 값(PH) 및 캐리 소모 비트 값(KH)을 계산한다. 도 3에서, AL<i> 및 BL<i> 각각은 입력되는 가산수(A)와 피가산수(B)의 비트 값 AH<i> 및 BH<i>의 반전 신호이다.

<110> 도 4는 도 1의 블록 캐리부(120)의 구체적인 회로도이다.

- <111> 도 4를 참조하면, 블록 캐리부(120)는, 합산 시작을 알리는 클럭 신호(ACLK)가 액티브될 때, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)을 상기 캐리 생성/전파부(110)에서 받아 블록 캐리들(K0, K4, 및 K8)을 생성한다.
- <112> 도 4에서, K0은 [논리식 1], 즉, 비트별 상기 캐리 소모 비트 값(KH)에 대한 제2 논리 조합에 의하여 하나 이상의 제2 논리 상태가 존재하면 제2 논리 상태로 된다. 그리고, K4는 [논리식 3], 즉, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들(도 2의 각 블록에서 하위 4비트)에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 된다. 또한, K8은 [논리식 5], 즉, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들(도 2의 각 블록에서 상위 4비트)에 제2 논리 상태인 상기 캐리 소모 비트 값(KH)이 하나 이상 존재하면 제2 논리 상태로 된다.
- <113> 도 5a 및 도 5b는 도 1의 비트 캐리부(130)의 구체적인 회로도이다. 도 5a 및 도 5b는, 도 5a의 오른쪽 끝 ??, ??, ??, ??, 및 ?? 각각이, 도 5b의 왼쪽 ??, ??, ??, ??, 및 ??에 연결되는, 하나의 회로도이다.
- <114> 도 5a 및 도 5b를 참조하면, 비트 캐리부(130)는, 합산 시작을 알리는 클럭 신호(ACLK)가 액티브될 때, 비트별 상기 캐리 전파 비트 값(PH) 및 비트별 상기 캐리 소모 비트 값(KH)을 받아, 비트별 캐리들(C0, 및 C1)을 계산한다.
- <115> 도 5a 및 도 5b의 맨체스터 캐리 체인에서,  $C1_{<i>}$ 는 [논리식 6], 즉, 상기 캐리 전파 비트 값(PH) 및 상기 캐리 소모 비트 값(KH)에 대한 제5 논리 조합에 의하여 계산되는 비트별 캐리들(C0, 및 C1)로서, 첫 번째 비트에 캐리가 있는 경우에 해당한다. 또

한,  $C0_{i+1}$ 는 [논리식 6], 즉, 상기 캐리 전파 비트 값(PH) 및 상기 캐리 소모 비트 값(KH)에 대한 제5 논리 조합에 의하여 계산되는 비트별 캐리들( $C0$ , 및  $C1$ )로서, 첫 번째 비트에 캐리가 없는 경우에 해당한다.

<116> 도 6은 도 1의 블록 캐리 선택부(140)의 구체적인 회로도이다.

<117> 도 6을 참조하면, 블록 캐리 선택부(140)는, 합산 시작을 알리는 클럭 신호(ACLK)가 액티브될 때, 상기 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ ) 및 하위 비트 블록으로부터 전파되는 캐리 값에 대한 소정 논리 조합으로 만들어지는 INL 신호에 의하여, 제1 논리 상태 또는 제2 논리 상태를 가지며, 어떠한 클럭 신호에 의하여도 출력을 제어 받지 않는 블록 캐리 발생 정보(CINH, 및 CINL)를 생성하는 동기화기(synchronizer)이다. 여기서, 하위 비트 블록은, 상기 합산이 이루어지는 블록의 바로 이전 하위 비트 블록, 및 상기 합산이 이루어지는 블록의 2번째 이전 하위 비트 블록을 포함한다. 즉, 하위 블록에서의 캐리 전파는 바로 이전 하위 비트 블록 또는 2번째 이전 하위 비트 블록에서 발생할 수 있고, 2번째 이전 하위 비트 블록에서 캐리가 전파되는 것은, 바로 이전 하위 비트 블록과 2번째 이전 하위 비트 블록 모두에서 캐리가 발생된 경우에 발생한다. 도 6에서, ADDEN 신호는 합산 동작을 할 수 있도록 하는 인에이블(enable) 신호로서, 합산 동작시 제1 논리 상태로 되며, INL 신호는 클럭 신호(ACLK)가 비활성화 상태, 즉, 제1 논리 상태일 때, 제2 논리 상태로 프리차징(precharging)된다.

<118> 여기서, 상기 블록 캐리 발생 정보(CINH, 및 CINL)는, 서로 반전된 논리 상태를 가지는 한 쌍의 디지털 정보(CINH, CINL)이다. 상기 블록 캐리들( $K0$ ,  $K4$ , 및  $K8$ ) 모두 제1 논리 상태로서 블록 캐리가 있는 경우에는, INL 신호가 제1 논리 상태로 되고, 이에 따라 발생하는 제2 논리 상태의 CINH, 및 제1 논리 상태의 CINL에 의하여, 상기 합산 값

선택부(150)에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택된다. 또한, 상기 하위 비트 블록으로부터 전파되는 캐리가 있는 경우에도, INL 신호가 제1 논리 상태로 되고, 이에 따라 발생하는 제2 논리 상태의 CINH, 및 제1 논리 상태의 CINL에 의하여, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리(C1)가 선택된다.

<119> 마찬가지로, 상기 블록 캐리들(K0, K4, 및 K8) 중 어느 하나가 제2 논리 상태로서 블록 캐리가 없는 경우에, INL 신호는 프리차징(precharging)된 제2 논리 상태를 유지하며, 이에 따라 발생하는 제1 논리 상태의 CINH, 및 제2 논리 상태의 CINL에 의하여, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 없는 경우의 상기 비트별 캐리(C0)가 선택된다. 또한, 상기 하위 비트 블록으로부터 전파되는 캐리가 없는 경우에도, INL 신호는 프리차징(precharging)된 제2 논리 상태를 유지하며, 이에 따라 발생하는 제1 논리 상태의 CINH, 및 제2 논리 상태의 CINL에 의하여, 상기 합산 값 선택부(150)에서 상기 첫 번째 비트에 캐리가 없는 경우의 상기 비트별 캐리(C0)가 선택된다.

<120> 여기서, 블록 캐리 선택부(140)는 블록 캐리 발생 정보(CINH, 및 CINL)를 생성하고, 이에 따라 동기화(synchronizing)되는 상기 합산 값 선택부(150)는, 상기 블록 캐리들(K0, K4, 및 K8)이 생성되는 즉시, 같은 페이즈(phase)의 ACLK 내에서 최종 합산 결과(SUM)를 출력한다. 이와는 대조적으로, 종래의 캐리 록어헤더 애더(CLA)에서는, 클럭 신호(ACLK)가 액티브된 후에 블록 캐리들(K0, K4, 및 K8)의 생성을 위한 시간적 마진을 고려하여, 액티브된 클럭 신호(ACLK)의 한 페이즈(phase) 후에 감지 증폭 플립플롭(sense amp. F/F)이 구동되도록 한다. 이에 따라, 종래의 캐리 록어헤더 애더(CLA)에서는, 감지 증폭 플립플롭(sense amp. F/F)에 클럭 신호(ACLK)의 반전 신호(adder bar clock)가 인

가되므로, 한 페이즈(phase) 뒤진 최종 합산 결과(SUM)가 출력된다. 따라서, 위와 같은 본 발명의 일실시예에 따른 캐리 룩어헤더 애더(CLA)의 블록 캐리 선택부(140)는 고속 합산 동작을 가능하게 한다.

<121> 도 7은 도 1의 합산 값 선택부(150)의 구체적인 회로도들이다.

<122> 도 7을 참조하면, 합산 값 선택부(150)는, 합산 시작을 알리는 클럭 신호(ACLK)가 액티브되고, 인에이블 신호 ENH 및 ENL 각각이 제2 논리 상태 및 제1 논리 상태로 될 때, 상기 블록 캐리 발생 정보(CINH, 및 CINL)에 대응하는 상기 비트별 캐리(C0, 또는 C1)을 선택하고, 선택된 상기 비트별 캐리(C0, 또는 C1)에 비트별 상기 캐리 전파 비트 값(PH)을 비트별로 합산하여 최종 캐리(CM)를 포함하는 최종 합산 결과(SUMH<i>, 및 SUMML<i>를 출력한다. 여기서, 상기 합산은, [표 2]와 같이, 상기 비트별 캐리(C0, 또는 C1)와 비트별 상기 캐리 전파 비트 값(PH)을 비트별로 XOR 로직 처리함으로써 계산된다.

<123> 도 8은 도 3 내지 도 7을 결합한 전체 회로에 대한 시뮬레이션 파형도이다.

<124> 도 8을 참조하면, 파형 1 내지 파형 5는 본 발명의 일실시예에 따라 도 3 내지 도 7을 결합한 캐리 룩어헤더 애더(CLA) 전체 회로의 시뮬레이션 파형이고, 파형 6 내지 파형 8은 종래의 캐리 룩어헤더 애더(CLA)의 시뮬레이션 파형이다.

<125> 도 8에서, 파형 1은 가산수 및 피가산수가 입력되는 시간(augend input)과 합산 시작을 알리는 클럭 신호(ACLK)가 액티브 상태로 되는 시간(adder clock)을 보여준다. 도 6의 ADDEN 신호는 클럭 신호(ACLK)가 반전된 신호이다. 파형 2는 가산수의 비트 값이 "0"이고, 피가산수의 비트 값이 "1"일 때와 같이, 캐리 전파 비트 값(PH)이 제2 논리 상태로 되는 것(propagation term)을 보여준다. 파형 3은 비트 캐리부(130)에서 생성되는

비트 캐리들(bit carry)을 보여주며, 맨체스터 캐리 체인으로 구현된 비트 캐리부(130)에서는 상위 비트일수록 계산되는 딜레이가 커짐을 알 수 있다. 파형 4는 본 발명의 일 실시예에 따른 블록 캐리들(K0, K4, 및 K8)이 선택되어 나오는 것을 보여준다. 파형 5는 블록 캐리들(K0, K4, 및 K8)이 생성되자마자 즉시 최종 합산 결과(sum results)가 계산되는, 본 발명의 특징을 보여준다.

<126> 파형 6은 종래의 캐리 룩어헤더 애더(CLA)에서, 클럭 신호(ACLK)가 액티브 된 후 한 페이즈(phase) 후에 감지 증폭 플립플롭(sense amp. F/F)이 구동되도록 하기 위한, 감지 증폭 플립플롭(sense amp. F/F)에 인가되는 클럭 신호(ACLK)의 반전 신호(adder bar clock)를 보여준다. 파형 7은 종래의 캐리 룩어헤더 애더(CLA)에서 발생하는 블록 캐리들(K0, K4, 및 K8)을 보여준다. 파형 8은 종래의 캐리 룩어헤더 애더(CLA)의 최종 합산 결과(sum results)가 계산되는 시간을 보여준다.

<127> 도 8에 도시된 바와 같이, 본 발명의 일 실시예에 따른 최종 합산 결과(sum results)가 계산되는 시간을 보여주는 파형 6은, 종래의 캐리 룩어헤더 애더(CLA)의 최종 합산 결과(sum results)가 계산되는 시간을 보여주는 파형 8에 비하여 한 페이즈(phase) 정도 빠르게 출력된다. 즉, 본 발명의 일 실시예에 따른 캐리 룩어헤더 애더(CLA)는 블록 캐리부(120)가 블록 캐리들(K0, K4, 및 K8)을 생성하는 즉시 최종 합산 결과(sum results)를 출력한다.

<128> 위에서 기술한 바와 같이, 본 발명의 일 실시예에 따른 캐리 룩어헤드 애더(CLA)는, 먼저, 캐리 생성/전파부(110)가 N 비트의 바이너리 값으로 입력되는 가산수와 피가산수를 M 블록들로 분별하여, 상기 블록들 각각에서 비트별로 제1 논리 조합하여 캐리 전파 비트 값(PH) 및 캐리 소모 비트 값(KH)을 계산한다. 블록 캐리부(120)는 블록 캐리들

(K0, K4, 및 K8)을 생성하고, 맨체스터 캐리 체인으로 구성되는 비트 캐리부(130)는 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 비트별 캐리들(C0, 및 C1)을 계산한다.

<129> 이에 따라, 블록 캐리 선택부(140)가 블록 캐리 발생 정보(CINH, 및 CINL)를 생성하면, 블록 캐리 발생 정보(CINH, 및 CINL)에 동기화(synchronizing)된 합산 값 선택부(150)는 상기 블록 캐리 발생 정보(CINH, 및 CINL)에 대응하는 상기 비트별 캐리(C0, 또는 C1)을 선택하고, 선택된 상기 비트별 캐리(C0, 또는 C1)에 비트별 상기 캐리 전파 비트 값(PH)을 비트별로 합산하여 최종 캐리(CM)를 포함하는 최종 합산 결과(SUM)를 출력한다.

<130> 이상에서와 같이 도면과 명세서에서 최적 실시예가 개시되었다. 여기서 특정한 용어들이 사용되었으나, 이는 단지 본 발명을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 발명의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

#### 【발명의 효과】

<131> 상술한 바와 같이 본 발명에 따른 캐리 록어헤드 애더(CLA)는, 맨체스터 캐리 체인을 사용한 캐리 록어헤드 애더(CLA)로서, 블록 캐리가 생성되어 전파되는 즉시, 블록 캐리에 의하여 선택되는 비트 캐리로부터 최종 합산 값을 계산하는 소정 동기화기(synchronizer)를 가진다. 따라서, 감지 증폭 플립플롭(sense amp.

F/F)을 액티브시키는 인에이블 신호를 기다리는 시간적 손실을 없앨 수 있기 때문에, 바이너리 값을 가지는 가산수와 피가산수에 대하여, 클럭 신호의 한 페이즈 내에 고속 합산을 수행 할 수 있는 효과가 있다.

## 【특허청구범위】

## 【청구항 1】

N 비트의 바이너리 값으로 입력되는 가산수와 피가산수를 M 블록들로 분별하여, 상기 블록들 각각에서 비트별로 제1 논리 조합하여 캐리 전파 비트 값 및 캐리 소모 비트 값을 계산하는 캐리 생성/전파부;

상기 블록들 각각에서, 비트별 상기 캐리 소모 비트 값에 대한 제2 논리 조합에 의하여 하나 이상의 제2 논리 상태가 존재하면 제2 논리 상태로 되는 K0, 비트별 상기 캐리 전파 비트 값 및 비트별 상기 캐리 소모 비트 값에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값이 하나 이상 존재하면 제2 논리 상태로 되는 K4, 및 비트별 상기 캐리 전파 비트 값 및 비트별 상기 캐리 소모 비트 값에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값이 하나 이상 존재하면 제2 논리 상태로 되는 K8로 이루어지는 블록 캐리들을 생성하는 블록 캐리부;

상기 블록들 각각에서, 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 상기 캐리 전파 비트 값 및 상기 캐리 소모 비트 값에 대한 제5 논리 조합으로 비트별 캐리들을 맨체스터 캐리 체인에 의하여 계산하는 비트 캐리부;

상기 블록들 각각에서, 상기 블록 캐리들 및 하위 비트 블록으로부터 전파되는 캐리 값에 대한 소정 논리 조합에 의하여, 제1 논리 상태 또는 제2 논리 상태를 가지며, 어떠한 클럭 신호에 의하여도 출력을 제어 받지 않는 블록 캐리 발생 정보를 생성하는 블록 캐리 선택부; 및

상기 블록 캐리 발생 정보에 대응하는 상기 비트별 캐리를 선택하고, 선택된 상기 비트별 캐리에 비트별 상기 캐리 전과 비트 값을 비트별로 합산하여 최종 합산 결과를 출력하는 합산 값 선택부를 구비하는 것을 특징으로 하는 캐리 록어헤드 애더.

**【청구항 2】**

제 1항에 있어서, 상기 블록 캐리 발생 정보는,

서로 반전된 논리 상태를 가지는 한 쌍의 디지털 정보이며, 상기 블록 캐리들 모두 제1 논리 상태로서 블록 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리가 선택되도록 하는 것을 특징으로 하는 캐리 록어헤드 애더.

**【청구항 3】**

제 2항에 있어서, 상기 블록 캐리 발생 정보는,

상기 하위 비트 블록으로부터 전파되는 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리가 선택되도록 하는 것을 특징으로 하는 캐리 록어헤드 애더.

**【청구항 4】**

제 1항 또는 제 3항에 있어서, 상기 하위 비트 블록은,

상기 합산이 이루어지는 블록의 바로 이전 하위 비트 블록, 및 상기 합산이 이루어지는 블록의 2번째 이전 하위 비트 블록을 포함하는 것을 특징으로 하는 캐리 록어헤드 애더.

## 【청구항 5】

제 1항에 있어서, 상기 제1 논리 조합은,

상기 가산수와 상기 피가산수에 대한 비트별 XOR 로직에 의하여 상기 캐리 전과 비트 값을 계산하는 로직, 및 상기 가산수와 상기 피가산수에 대한 비트별 NOR 로직에 의하여 상기 캐리 소모 비트 값을 계산하는 로직인 것을 특징으로 하는 캐리 룩어헤드 애더.

## 【청구항 6】

제 1항에 있어서, 상기 제2 논리 조합은,

비트별 상기 캐리 소모 비트 값에 대한 OR 로직인 것을 특징으로 하는 캐리 룩어헤드 애더.

## 【청구항 7】

제 1항에 있어서, 상기 제3 논리 조합은,

논리식 ,

$K4 =$

$PH <\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{N/(2M)+1\}>PH<N/(2M)>$

$\{KH<\{N/(2M)-1\}> +$

$PH <\{N/(2M)-1\}>KH<\{N/(2M)-2\}> +$

$PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>KH<\{N/(2M)-3\}> +$

$\dots + PH <\{N/(2M)-1\}>PH<\{N/(2M)-2\}>\dots PH<1>KH<0> +$

$$PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>\dots PH<1>PH<0>\}$$

( 여기서, K4는 하위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH<>$ 는 캐리 전파 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

을 만족하는 것을 특징으로 하는 캐리 룩어헤드 애더.

#### 【청구항 8】

제 1항에 있어서, 상기 제4 논리 조합은,

논리식 ,

$$K8 =$$

$$KH <\{(N/M)-1\}> +$$

$$PH<\{(N/M)-1\}>KH<\{(N/M)-2\}> +$$

$$\dots + PH <\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{(N/(2M)+2)\}>KH<\{N/(2M)+1\}> +$$

$$PH<\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{(N/(2M)+1)\}>KH<N/(2M)>$$

( 여기서, K8은 상위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH<>$ 는 캐리 전파 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

을 만족하는 것을 특징으로 하는 캐리 룩어헤드 애더.

#### 【청구항 9】

제 1항에 있어서, 상기 제5 논리 조합은,

논리식 ,

$$C0<i> = KH<i-1> + PH<i-1>C0<i-1>$$

$$C1 <i> = KH<i-1> + PH<i-1>C1<i-1>$$

(여기서,  $i$ 는 0부터  $(N/M-1)$ 까지,  $C1<i>$ 은 첫 번째 비트에 캐리가 있는 경우의 비트별 캐리들,  $C0<i>$ 은 첫 번째 비트에 캐리가 없는 경우의 비트별 캐리들,  $PH<>$ 는 캐리 전과 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

을 만족하는 것을 특징으로 하는 캐리 룩어헤드 애더.

#### 【청구항 10】

제 1항에 있어서, 상기 합산은,

상기 비트별 캐리들과 비트별 상기 캐리 전과 비트 값을 비트별로 XOR 로지 처리하여 계산하는 것을 특징으로 하는 캐리 룩어헤드 애더.

#### 【청구항 11】

$N$  비트의 바이너리 값으로 입력되는 가산수와 피가산수를  $M$  블록들로 분별하여 계산되는 블록 캐리들에 대응하여, 상기 블록들 각각의 맨체스터 캐리 체인에서 발생하는 비트 캐리들이 선택되도록 하는 캐리 룩어헤드 애더의 합산 방법에 있어서,

상기 캐리 룩어헤드 애더에 의하여, 상기 블록들 각각에서 상기 가산수와 상기 피가산수의 비트별로 제1 논리 조합하여 캐리 전과 비트 값 및 캐리 소모 비트 값을 계산하는 단계;

상기 캐리 룩어헤드 애더에 의하여, 상기 블록들 각각에서, 비트별 상기 캐리 소모 비트 값에 대한 제2 논리 조합에 의하여 하나 이상의 제2 논리 상태가 존재하면 제2 논리 상태로 되는  $K0$ , 비트별 상기 캐리 전과 비트 값 및 비트별 상기 캐리 소모 비트 값에 대한 제3 논리 조합에 의하여 하위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모

비트 값이 하나 이상 존재하면 제2 논리 상태로 되는 K4, 및 비트별 상기 캐리 전파 비트 값 및 비트별 상기 캐리 소모 비트 값에 대한 제4 논리 조합에 의하여 상위  $N/(2M)$  비트들에 제2 논리 상태인 상기 캐리 소모 비트 값이 하나 이상 존재하면 제2 논리 상태로 되는 K8로 이루어지는 블록 캐리들을 생성하는 단계;

상기 캐리 룩어헤드 애더에 의하여, 상기 블록들 각각에서, 첫 번째 비트에 캐리가 있는 경우와 없는 경우 각각에 대하여, 상기 캐리 전파 비트 값 및 상기 캐리 소모 비트 값에 대한 제5 논리 조합으로 상기 비트별 캐리들을 상기 맨체스터 캐리 체인에 의하여 계산하는 단계;

상기 캐리 룩어헤드 애더에 의하여, 상기 블록들 각각에서, 상기 블록 캐리들 및 하위 비트 블록으로부터 전파되는 캐리 값에 대한 소정 논리 조합에 의하여, 제1 논리 상태 또는 제2 논리 상태를 가지며, 어떠한 클럭 신호에 의하여도 출력을 제어 받지 않는 블록 캐리 발생 정보를 생성하는 단계; 및

상기 캐리 룩어헤드 애더에 의하여, 상기 블록 캐리 발생 정보에 대응하는 상기 비트별 캐리를 선택하고, 선택된 상기 비트별 캐리에 비트별 상기 캐리 전파 비트 값을 비트별로 합산하여 최종 합산 결과를 출력하는 단계를 구비하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

#### 【청구항 12】

제 11항에 있어서, 상기 블록 캐리 발생 정보는,

서로 반전된 논리 상태를 가지는 한 쌍의 디지털 정보이며, 상기 블록 캐리들 모두 제1 논리 상태로서 블록 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비

트에 캐리가 있는 경우의 상기 비트별 캐리가 선택되도록 하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

#### 【청구항 13】

제 12항에 있어서, 상기 블록 캐리 발생 정보는,

상기 하위 비트 블록으로부터 전파되는 캐리가 있는 경우에, 상기 합산 값 선택부에서 상기 첫 번째 비트에 캐리가 있는 경우의 상기 비트별 캐리가 선택되도록 하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

#### 【청구항 14】

제 11항 또는 제 13항에 있어서, 상기 하위 비트 블록은,

상기 합산이 이루어지는 블록의 바로 이전 하위 비트 블록, 및 상기 합산이 이루어지는 블록의 2번째 이전 하위 비트 블록을 포함하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

#### 【청구항 15】

제 11항에 있어서, 상기 제1 논리 조합은,

상기 가산수와 상기 피가산수에 대한 비트별 XOR 로직에 의하여 상기 캐리 전파 비트 값을 계산하는 로직, 및 상기 가산수와 상기 피가산수에 대한 비트별 NOR 로직에 의하여 상기 캐리 소모 비트 값을 계산하는 로직인 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

#### 【청구항 16】

제 11항에 있어서, 상기 제2 논리 조합은,

비트별 상기 캐리 소모 비트 값에 대한 OR 로직인 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

【청구항 17】

제 11항에 있어서, 상기 제3 논리 조합은,

논리식 ,

$K4 =$

$PH<\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{N/(2M)+1\}>PH<N/(2M)>$

$\{KH<\{N/(2M)-1\}> +$

$PH<\{N/(2M)-1\}>KH<\{N/(2M)-2\}> +$

$PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>KH<\{N/(2M)-3\}> +$

$\dots + PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>\dots PH<1\>KH<0\> +$

$PH<\{N/(2M)-1\}>PH<\{N/(2M)-2\}>\dots PH<1\>PH<0\>\}$

( 여기서, K4는 하위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH<>$ 는 캐리 전파 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

을 만족하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

【청구항 18】

제 11항에 있어서, 상기 제4 논리 조합은,

논리식,

$K8 =$

$$KH<\{(N/M)-1\}> +$$

$$PH <\{(N/M)-1\}>KH<\{(N/M)-2\}> +$$

$$\dots + PH<\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{(N/(2M)+2)\}>KH<\{(N/(2M)+1)\}> +$$

$$PH <\{(N/M)-1\}>PH<\{(N/M)-2\}>\dots PH<\{(N/(2M)+1)\}>KH<N/(2M)>$$

(여기서,  $K8$ 은 상위  $N/(2M)$  비트들에 캐리 소모 비트가 발생하는지 체크하는 블록 캐리,  $PH<>$ 는 캐리 전파 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

을 만족하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

#### 【청구항 19】

제 11항에 있어서, 상기 제5 논리 조합은,

논리식 ,

$$C0<i> = KH<i-1> + PH<i-1>C0<i-1>$$

$$C1 <i> = KH<i-1> + PH<i-1>C1<i-1>$$

(여기서,  $i$ 는 0부터  $(N/M-1)$ 까지,  $C1<i>$ 은 첫 번째 비트에 캐리가 있는 경우의 비트별 캐리들,  $C0<i>$ 은 첫 번째 비트에 캐리가 없는 경우의 비트별 캐리들,  $PH<>$ 는 캐리 전파 비트 값,  $KH<>$ 는 캐리 소모 비트 값)

을 만족하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

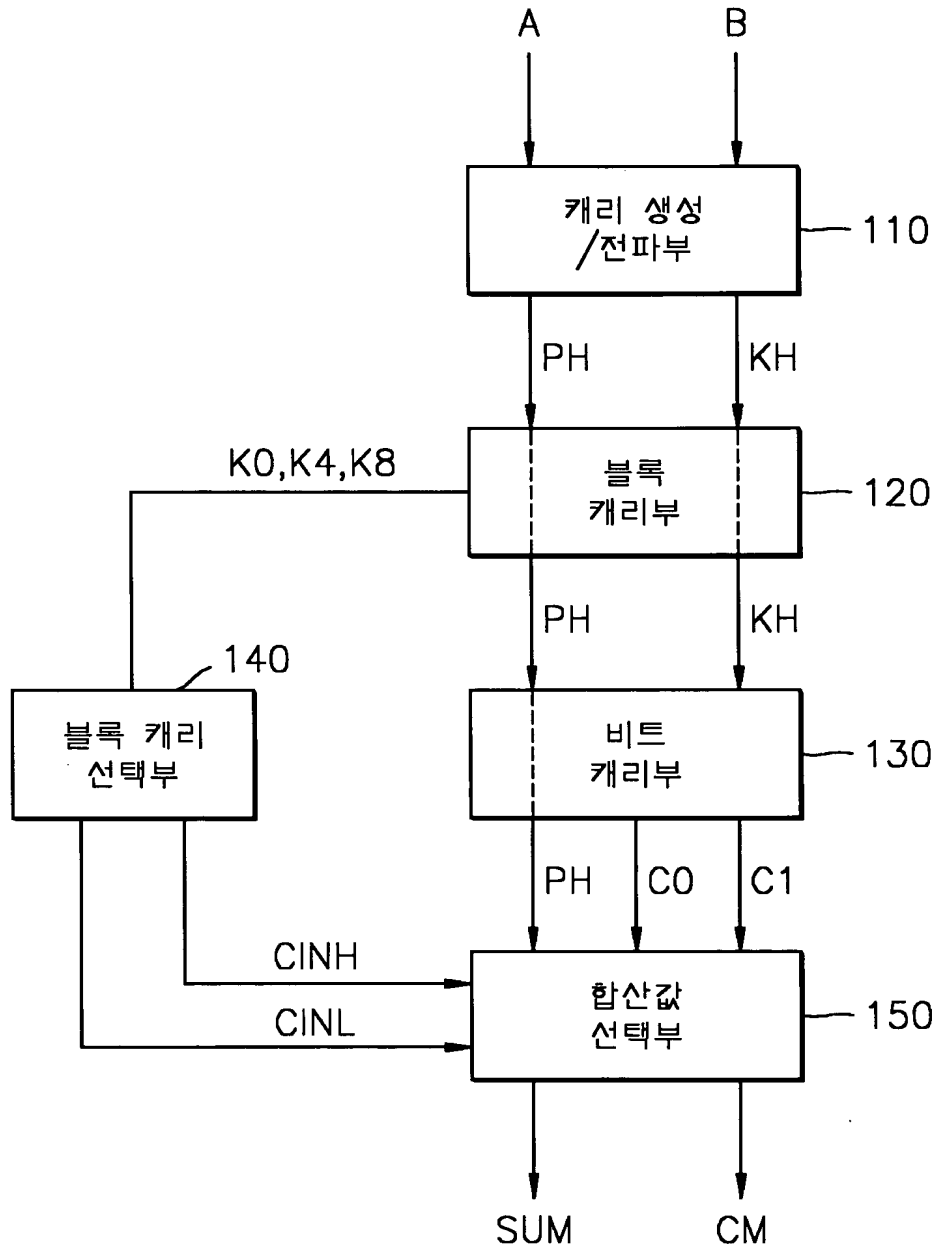
#### 【청구항 20】

제 11항에 있어서, 상기 합산은,

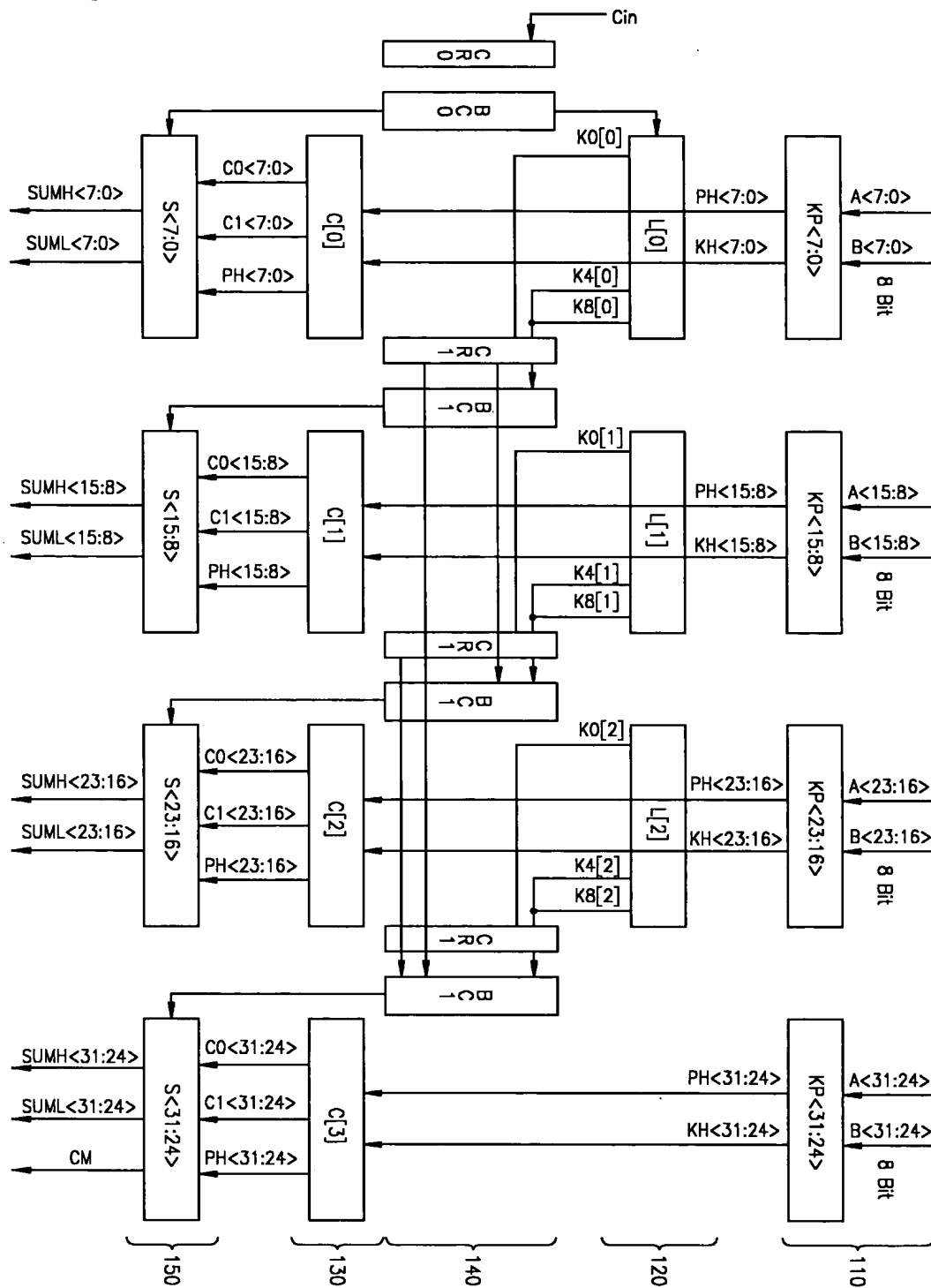
상기 비트별 캐리들과 비트별 상기 캐리 전파 비트 값을 비트별로 XOR 로직 처리하여 계산하는 것을 특징으로 하는 캐리 룩어헤드 애더의 합산 방법.

## 【도면】

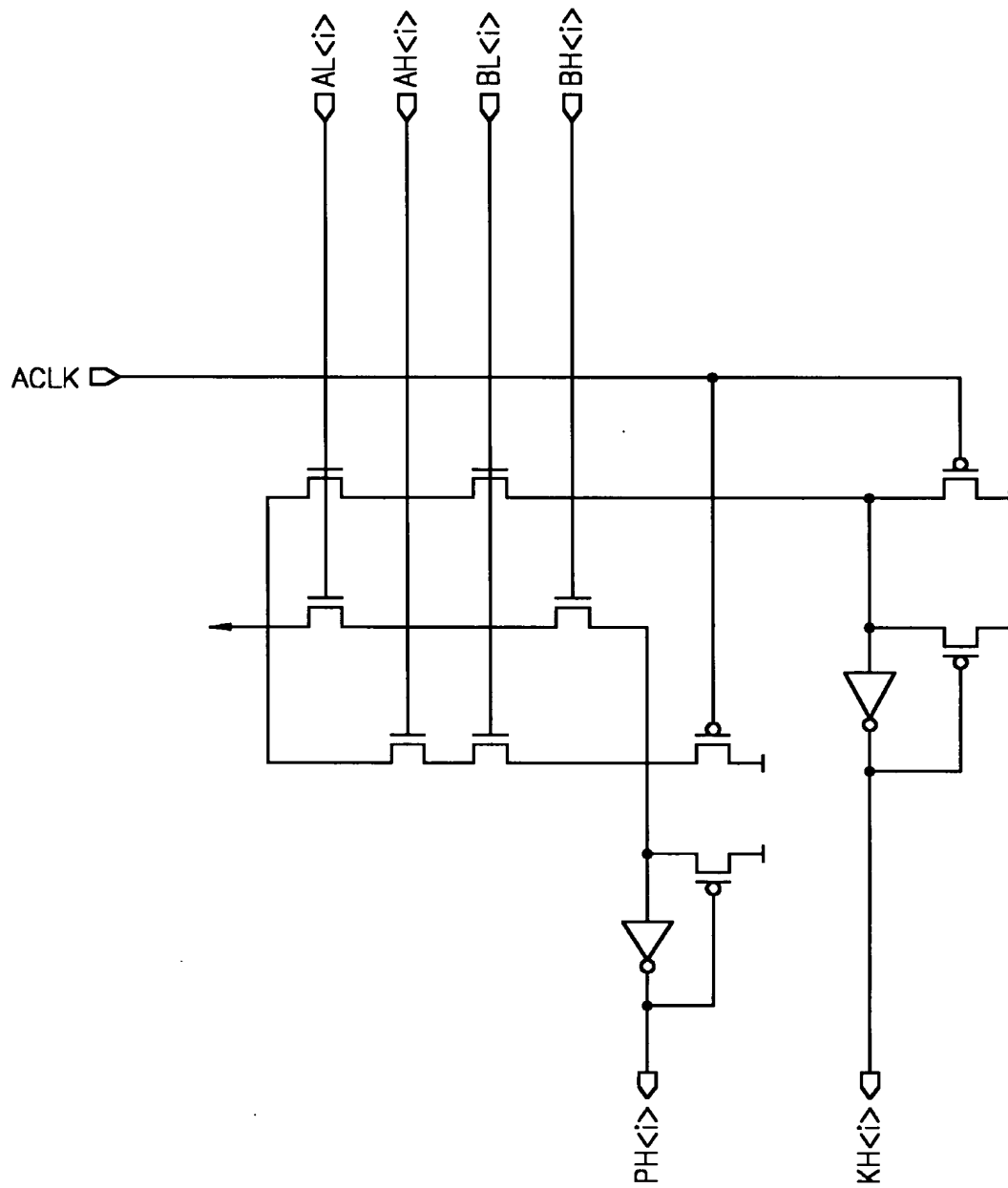
【도 1】



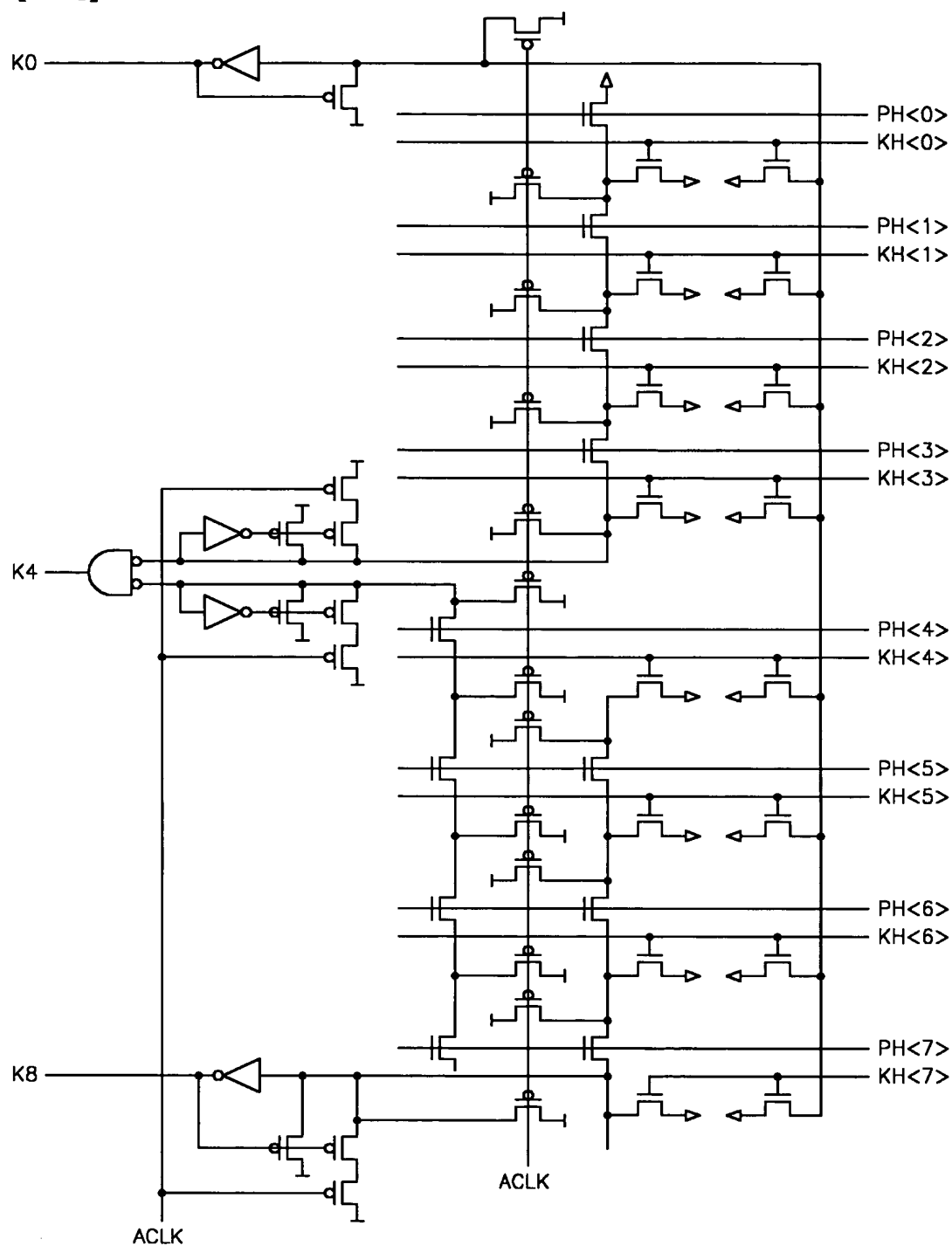
【도 2】



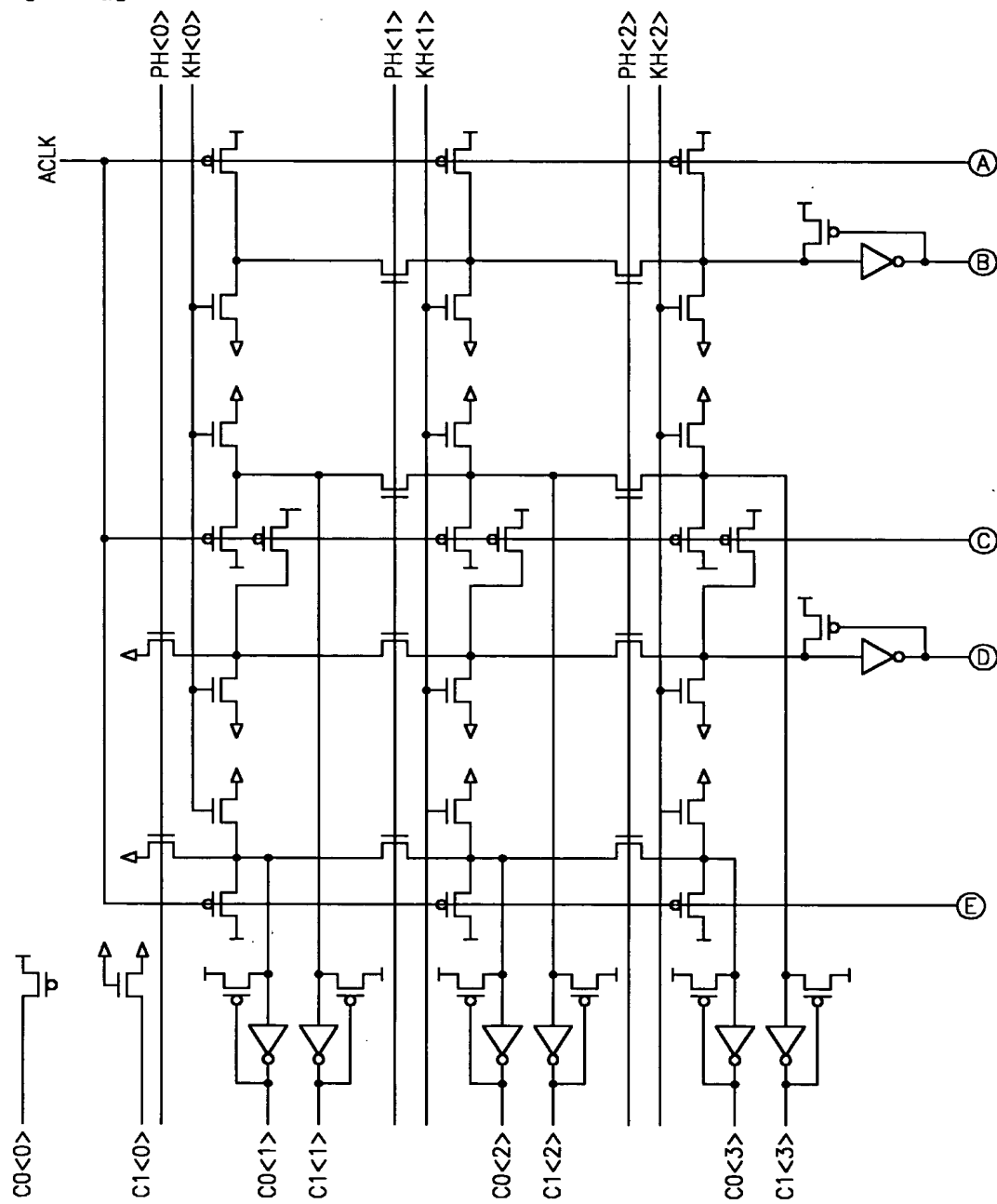
【도 3】



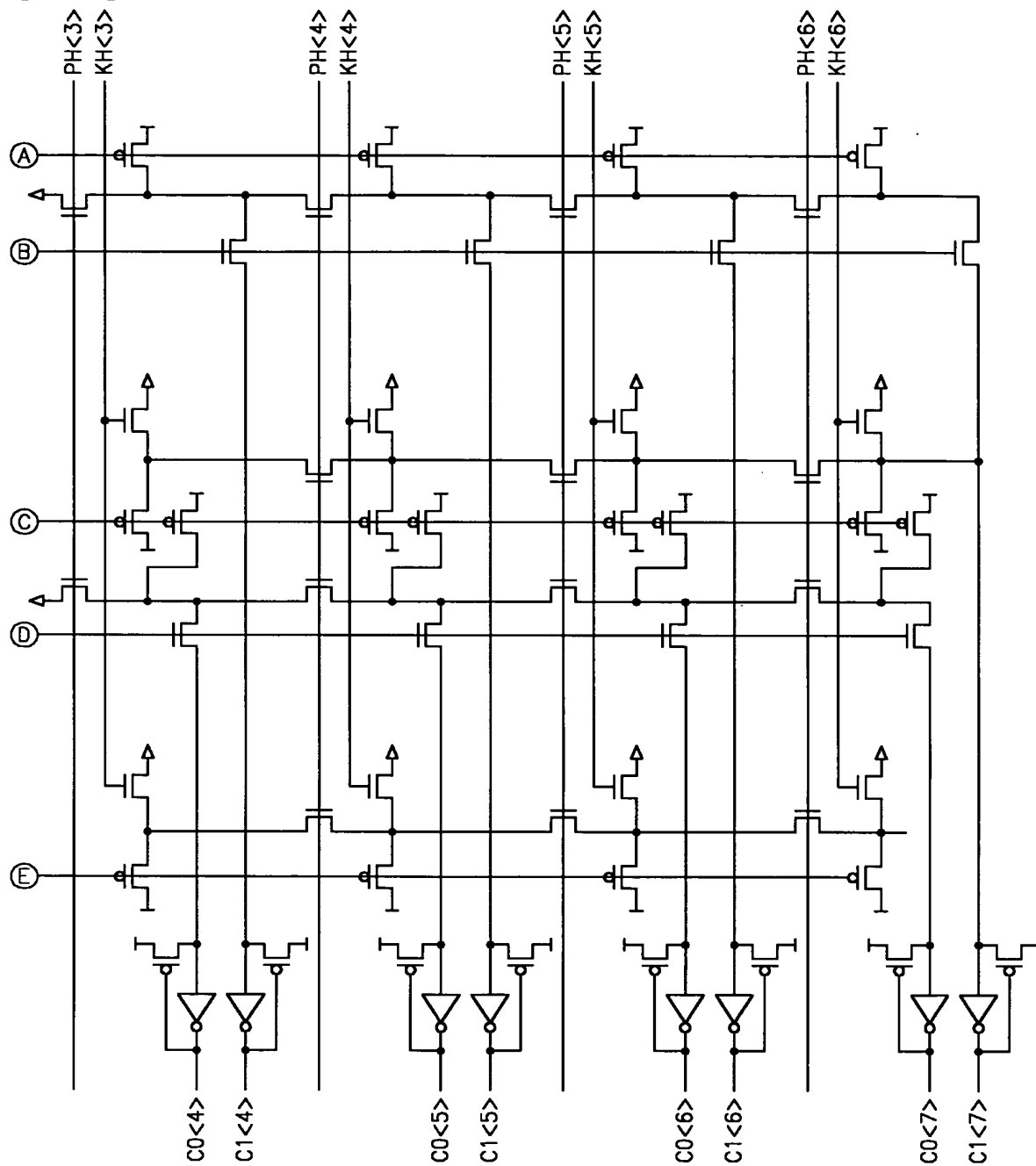
【도 4】



【도 5a】



【도 5b】

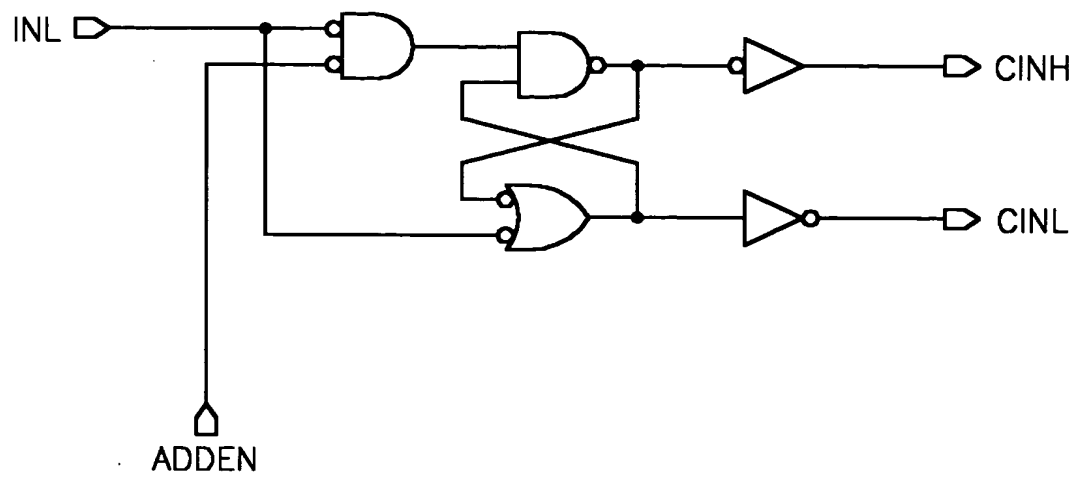




1020030011210

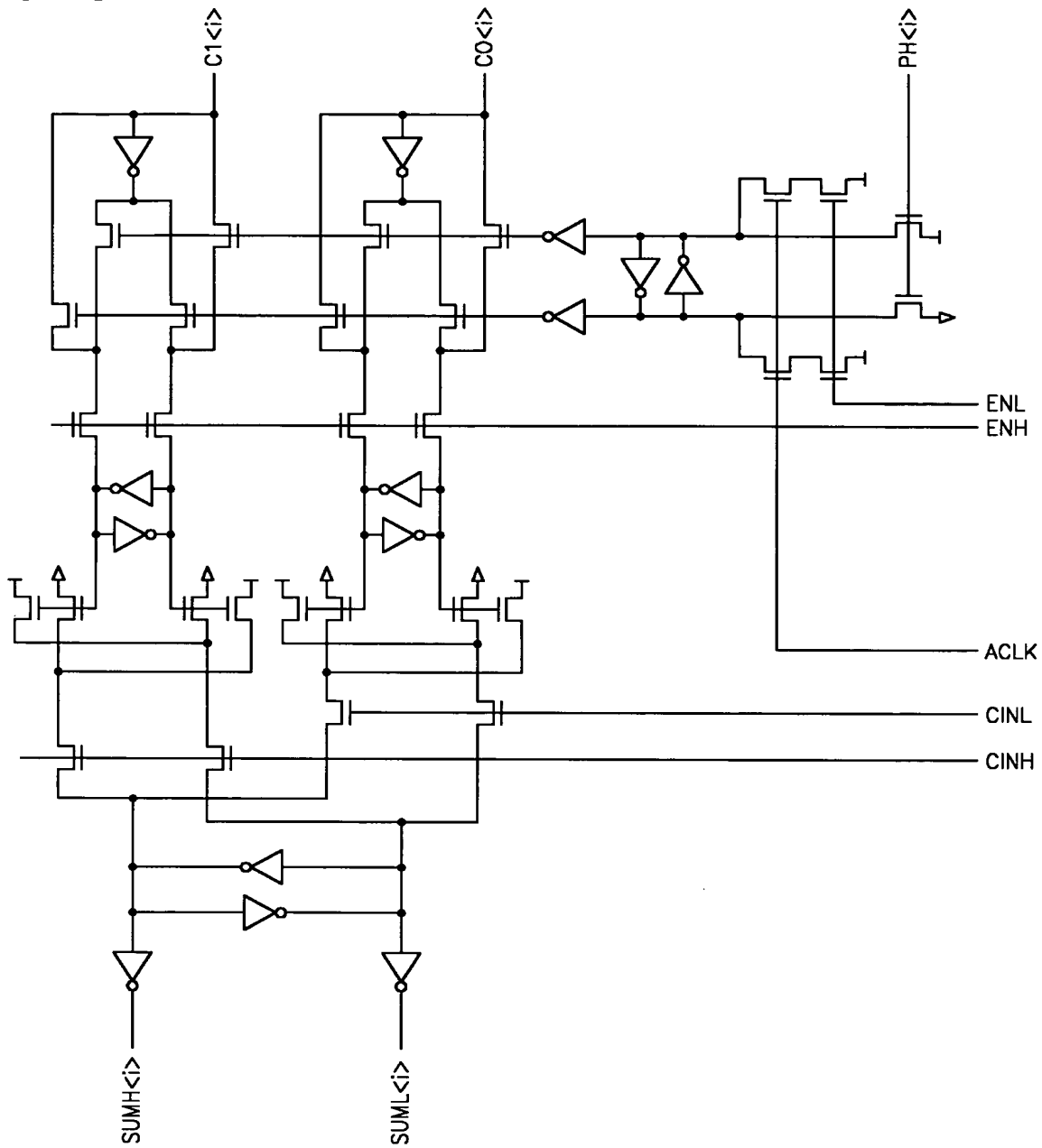
출력 일자: 2003/7/16

【도 6】





【도 7】



【도 8】

